

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-296302  
 (43)Date of publication of application : 17.10.2003

(51)Int.Cl. G06F 17/16  
 H03M 13/09  
 H03M 13/19

(21)Application number : 2002-098025  
 (22)Date of filing : 29.03.2002

(71)Applicant : FUJITSU LTD  
 (72)Inventor : OTA MITSUHIKO

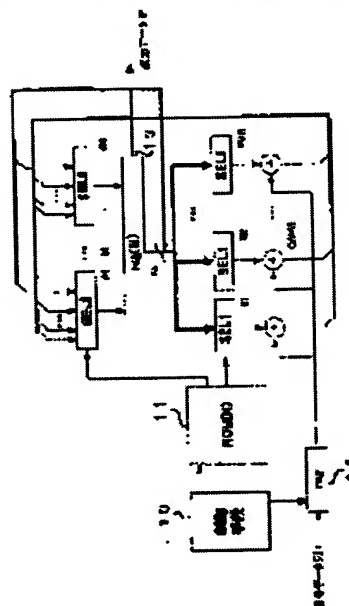
## (54) MATRIX ARITHMETIC PROCESSOR

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a matrix arithmetic processor capable of quickly executing a matrix arithmetic operation in a small circuit scale.

**SOLUTION:** An input signal data column I is temporarily stored in a reg 12, and inputted to an adder in response to an instruction from a control means 10. The control means 10 instructs an ROM 11 in which a check matrix H is stored to obtain a position where '1' is erected in a certain column of the check matrix. The ROM 11 instructs selectors SEL1 #1 to #CW to select values corresponding to the position of 1 in the check matrix from among values from a reg(M) 13, and to transmit the values to the adder. The added values are selected by selectors SEL2 instructed by the ROM 11, and inputted to the reg(M) 13. As for non-added values, the output values of the reg(M) 13 are inputted as they are, or inputted through the selectors SEL2 to the reg(M). This process is repeated until all the arithmetic operations end.

本発明の実施形態に基いた行列演算回路の構成例



## LEGAL STATUS

[Date of request for examination] 10.12.2004  
 [Date of sending the examiner's decision of rejection]  
 [Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]  
 [Date of final disposal for application]  
 [Patent number]  
 [Date of registration]  
 [Number of appeal against examiner's decision of rejection]  
 [Date of requesting appeal against examiner's decision of rejection]  
 [Date of extinction of right]



## 【特許請求の範囲】

【請求項 1】 行列の行列要素を格納する格納手段と、初期値として全て 0 である値を格納し、順次行われる演算結果を順次格納するレジスタ手段と、該レジスタ手段から出力される値と入力データ値とを加算する加算手段と、前記行列の行列要素値に基づいて、必要な値を該レジスタ手段から該加算手段に入力し、入力データ値と該レジスタ手段からの値を加算させる演算制御手段と、加算器の出力と、該レジスタ手段の出力を適切に選択して再び該レジスタ手段に格納するループバック手段と、を備える行列演算処理装置。

【請求項 2】 前記格納手段は、検査行列の行列要素が 1 となる行列要素の位置のみを情報として格納することを特徴とする請求項 1 に記載の行列演算処理装置。

【請求項 3】 前記演算制御手段は、入力データ値の入力順に合わせて前記格納手段から必要な情報を出力させることを特徴とする請求項 1 に記載の行列演算処理装置。

【請求項 4】 前記行列は、データの符号化に伴う誤り訂正符号の検査行列であることを特徴とする請求項 1 に記載の行列演算処理装置。

【請求項 5】 前記行列は、LDPC (Low Density Pa

$$I = [i_0, i_1, \Lambda, i_{N-1}], P = [p_0, p_1, \Lambda, p_{M-1}], H = \begin{bmatrix} h_{00} & h_{01} & \Lambda & h_{0N-1} \\ M & M & M & M \\ h_{M-10} & h_{M-11} & \Lambda & h_{M-1N-1} \end{bmatrix}$$

【0005】とした場合、

【0006】

【数 2】

$$P_m = \sum_{n=0}^{N-1} h_{mn} * i_n \quad \dots\dots\dots(1)$$

【0007】となる。例えば、記録再生装置の代表例である磁気ディスク装置に関して、エラー訂正機能が搭載され、その符号の一つとして LDPC 符号があり、この符号にも行列演算が必須となっている。

【0008】ここで、パリティ計算や LDPC 復号で用いられる検査行列は通常、行列要素が 1 あるいは 0 であるので、行列中要素が 1 である項を加算することになり、(1) 式は、

【0009】

【数 3】

$$P_m = \sum_{n=0/h_{mn}=1}^{N-1} i_n \quad \dots\dots\dots(2)$$

【0010】となる。

【0011】

【発明が解決しようとする課題】 図 7 は、従来の行列演算回路の例である。従来、上記の処理を行い処理結果 P を得るためには、信号データ列 I の全データを取得後、

arity Check) 符号の検査行列であることを特徴とする請求項 1 に記載の行列演算処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、行列演算処理装置に関する。

【0002】

【従来の技術】 現在、パソコンをはじめ多くの情報処理装置が実用化され、実際に使用されている。このような情報処理装置においては、データを格納したり、転送する場合、データを符号化して格納あるいは転送する。特に、デジタル信号を磁気ディスク、光ディスク、光磁気ディスクなどの記録再生媒体あるいはネットワークから受信して復号する場合に使用する誤り訂正として LDPC (Low Density Parity Check) 符号がある。

【0003】 エラー訂正などを行う場合、N ビットの信号データ列 I と N×M の検査行列 H を用いて処理結果 P を得るには、 $P = H * I^T$  という行列演算が必要となる。例えば、

【0004】

【数 1】

(2) 式の計算を行う必要がある。

【0012】 信号データ列 I は、レジスタ 40 にその全データが蓄積された後、行列データ H を格納した ROM 41 などから読み出された各行の中の 1 となっている項がセクタ SEL 42 によってセレクトされ、加算器 46 で加算される。加算結果は、レジスタ 43 に格納される。ここで RW は各行の 1 の最大個数とする。これを M 回繰り返すことで処理結果 P を得る。ここでは P の全データを得るまでセクタ/加算器を共用した場合でも、N+M 時間が必要となり遅延が大きいという問題がある。また、回路規模も大きな保持レジスタ N が必要であったり、RW 個の加算器が必要となる。

【0013】 図 8 は、別の従来例を示す図である。本従来例では、信号データ列 I はレジスタ 44 にその全データが蓄積された後、行列 H に従って結線された加算器 47 により処理結果 P の全データを計算する。結果はレジスタ 45 に格納され、出力される。この場合、P の全データを得るまで必要な時間は N であるが、保持レジスタ N が大きいと言うことと、RW×M 個の加算器が必要となり回路規模が大きいという問題がある。

【0014】 本発明の課題は、小さな回路規模で高速に行列演算することのできる行列演算処理装置を提供することである。

【0015】

【課題を解決するための手段】本発明の行列演算処理装置は、行列の行列要素を格納する格納手段と、初期値として全て0である値を格納し、順次行われる演算結果を順次格納するレジスタ手段と、該レジスタ手段から出力される値と入力データ値とを加算する加算手段と、前記行列の行列要素値に基づいて、必要な値を該レジスタ手段から該加算手段に入力し、入力データ値と該レジスタ手段からの値を加算させる演算制御手段と、加算器の出力と、該レジスタ手段の出力を適切に選択して再び該レジスタ手段に格納するループバック手段とを備える。

【0016】本発明によれば、従来に比べ、加算器が少なく、回路規模が小さくなる。また、行列の格納も必要な情報のみを格納するようにすれば、メモリ容量も小さくでき、小型かつ高速な行列演算処理装置を提供することができる。

【0017】

【発明の実施の形態】Nビットの信号データ列IとN×Mの検査行列Hを用いて処理結果Pを得る $P=H \times I^T$ という行列演算において、検査行列Hの各列単位で必要となる演算を行い、処理結果Pの各要素を行分累積加算することで処理遅延/回路規模を削減する。特に、符号化における誤り訂正符号の検査行列においては、検査行列の列数Mが行数Nよりもかなり小さくなるので、列方向に並列に計算し、行方向に累積演算するようにすることにより、加算器の数を減らし、回路規模を小さくすることができる。

【0018】レジスタなどの処理結果Pの保持手段、ROMなどの検査行列Hの保持手段、アドレスカウンタなどの信号データ列Iの受信により検査行列H及び処理結果Pの読み出し、格納を制御する手段、加算器などの演算手段を備え、信号データ列Iの受信毎に入力データの処理が必要となる列データをHから得て、対象となる処理結果Pの必要項を読み出し、受信データと演算、結果を処理結果Pの保持手段に書き戻す。これを信号データ列Iの全受信データに繰り返し行うことで、処理結果Pを得る。

【0019】図1は、本発明の実施形態に従った行列演算回路の構成例である。制御手段10は信号データ列Iを受信制御し、保持手段reg12に保持する。また、受信データ位置から検査行列Hの保持手段11より列の処理対象の処理結果Pの位置を得る。処理結果Pの格納手段reg(M)13はデータ受信前に全て0に初期化される。制御手段10と保持手段11によって処理対象の処理結果PはデータセクタSEL1で選択され、入力データと加算処理される。ここで加算器は列中の1の最大個数CW個。SEL1はM→CWのセクタでCW個、SEL2は(CW+1)→1のセクタでM個となる。加算結果及びreg(M)13より読み出されたデータは

SEL2により処理データか元データが選択されreg(M)13に書き戻される。

【0020】すなわち、図1においては、信号データ列Iはreg12に1ビット格納され、制御手段10によって呼び出され、各加算器に入れられる。制御手段は、ROM11の中から検査行列Hの適切な列の内、1となっている位置を検出し、reg(M)13から来た信号の内、検査行列の当該列内の1となっている位置に対応するセクタSEL1がreg(M)13からの値を選択し、信号データ列と加算させる。加算結果は、セクタSEL2に送られ、セクタSEL2では、加算が行われたデータ値については、加算器からの値をreg(M)13に入力し、加算が行われなかった値については、reg(M)13からの値を再び、reg(M)13に入力する。そして、全ての信号データ列Iの受信、演算が終わった時点でreg(M)13は、出力データPを出力する。

【0021】より概念的に述べれば、行列の縦方向を列方向、横方向を行方向とすると、入力信号データ列Iを読み込んだとき、Iの入力順を予め知っておくことによって、Iの要素をどの列に乗算すべきかを決定しておく。そして、Iが入力されると、演算すべき検査行列の列を読み出し、並列に列方向の演算をする（実際には、検査行列の要素は0か1であるので、行列要素が1である行の値をそのまま加算器に入力する）。そして、演算結果をreg(M)に格納し、Iが順次入力される毎に、行方向に加算していき、全ての加算が終わった時点でパリティの列ベクトルが得られ、演算終了とする手順である。

【0022】本装置によればデータの処理遅延はN、回路規模は加算器はCW個、保持レジスタはMとなる。検査行列Hの保持手段の例として、列中の1の位置のアドレス保持する場合、(Cadd1, Cadd2, Cadd3)をROM(H)の1番目のアドレスに格納する。ここでCaddnは列n番目の1の位置を表し、1の個数がCW個未満の場合は0とする。

【0023】例えば、H(N=8, M=5)を、

【0024】

【数4】

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

【0025】とした場合は、

【0026】

【表1】

アドレス	データ
0	(1, 3, 4)
1	(2, 4, 5)
2	(1, 3, 5)
3	(1, 2, 4)
4	(2, 4, 5)
5	(1, 2, 5)
6	(1, 2, 3)
7	(2, 3, 4)

【0027】となる。この場合各アドレスは3ビット(0～5<8)で表現可能なので、CW=3なら3×3=9ビットを1ワードとして格納すれば良い。入力データ列Iに応じ、0番目なら(1, 3, 4)が出力され、各SEL1は左から1, 3, 4のMを選択し、SEL2は左から1, 3, 4番目のSEL2が各SEL1からの信号を、それ以外のSEL2はMからの信号を選択する。

【0028】SEL1、SEL2の制御実現例を以下に示す。SEL1はM→1のセクタで、制御信号がm(整数)を示すものであるとすると、m番目のデータを

選択出力し、この場合ROMからの出力をMSBより3ビットずつ制御信号mとすれば良い。

【0029】SEL2は4(CW+M)→1のセクタで、制御信号は、0→M、1→SEL1#1、2→SEL1#2、4→SEL1#CWを選択するものとし、ROMからの出力を表2のデータマルチプレクサデコード入力、各デコードの出力を図Aのように制御信号とすることで実現できる。

20 【0030】

【表2】

入力	出力データ
0	0 (00000b)
1	1 (00001b)
2	2 (00010b)
3	4 (00100b)
4	8 (01000b)
5	16 (10000b)

【0031】図2は、本発明の実施形態の動作を示す回路例である。制御信号は、ROM(H)1から出力され、セクタSEL1#1～SEL1#CW(2-1～2-CW)に入力されると共に、マルチプレクサデコードDEC3-1～3-nそれぞれに入力される。マルチプレクサデコードDEC3-1～3-nからは、ROM(H)1からの制御信号をセクタSEL2の選択信号にデコードした信号が出力される。SEL2は、これらマルチプレクサデコードDEC3-1～3-nからの信号に基づいて制御される。H行列の読み出しアドレスの発生方法の例図3～図6は、本発明の実施形態に従った制御手段の実現例を示す図である。

【0032】まず、図3において、入力データ列Iの先頭においてはdata\_startが、また、データIが有効な期間はdata\_enableが図3のような場合、カウントインネーブル(“1”でカウントUP)、クリア(“1”で全て0)付きのカウント22を用いることで制御手段を構成可能である。ここで、前項の行列の場合は3ビットカウンタ(0～7)で可能である。

【0033】また、Mの保持レジスタは図3のようにdata\_enableが“1”の場合はSEL2出力を、“0”の場合はデータをセクタ20で選択し、FF21に保持することで実現できる。

【0034】ここで、data\_enableは図4のように不連続でも良い。入力データIが昇順で入力されない場合、降順であればHの格納を逆順で格納することで実現できる。

40 【0035】また、図4にてカウンタをロード機能付きダウンカウンタに変更、data\_startにて値N-1をロードし、data\_enableでダウンカウントすることでも実現可能である。

【0036】入力データIが予め定義された順番の場合は、Hの格納をその定義順で行うことで実現可能である。入力データIがインタリーブされた場合はHの格納をその順番で格納することで、実現可能である。

50 【0037】また、図5の構成でも可能である。nはインタリーブの間隔とする。ここでn=2とすると入力順は前項の例(0～7)では0、2、4、6、1、3、

5、7となる。

【0038】REG 33はアドレスを保持するFF、各セクタ31、32はdata\_enableが“1”の時、図5においてセクタ31、32の下側に入っているデータをセレクトする。比較器34は、カウンタ30のカウントインネブルと、セクタ31、32の上側のポートへの値の入力を行うものである。

【0039】各動作は、

1. data\_startでREGを全て0、カウンタ30はクリアで“1”をロードする。
2. data\_enableでREG 33にnを加算した値をロード（設定）。
3.  $REG > N - 1$ を比較、真なら4へ、偽なら2へ
4. data\_startなら1へ、それ以外はdata\_enableでREG 33へカウンタの値を設定、カウンタ30はCEで+1し、2へとなる。

【0040】本構成で $n = 1$ とすればインタリブ無しの昇順構成となる。また、図6の構成にし、

1. data\_startでREG 33を $N - 1$ 、カウンタ30はクリアで $N - 2$ をロードする。
2. data\_enableでREG 33にnを減算した値をロード（設定）。
3.  $REG = 0$ を比較、真なら4へ、偽なら2へ
4. data\_startなら1へ、それ以外はdata\_enableでREG 33へカウンタの値を設定、カウンタ30はCEで-1し2へ

とすることで逆順のインタリブ構成となる。

【0041】以上のような実施形態においては、LDPC符号における検査行列の一例で比較した場合、 $N = 4352$ 、 $M = 256$ 、 $CW = 3$ 、 $RW = 51$ とすると、本実施形態では、従来例に比較して保持メモリで $M/M = 1/4$ 、加算器は図7の構成と比較して $CW/RW = 3/51$ 、図8の構成と比較して、 $CW/RW \times M = 3/13056$ となる。

【0042】本実施形態の装置では処理は各列単位で行われる。これによりNビットの信号データ列I受信順序によらず、処理を行うことが可能となる。また、各列単位で処理を行うことでNビットの信号データ列Iの任意の位置から処理を行うことが可能となる。

【0043】更に、本実施形態においては検査行列Hの保持を処理が必要となる項のアドレスのみを保持することで保持手段の回路規模が削減可能となる。行列をそのまま保持した場合は、 $M$ ビット $\times N$ アドレス分の容量となるが、アドレスで保持した場合、 $\log_2(M)$ ビット $\times CW$ の容量となる。

【0044】また、信号データ列Iの受信順序の保存を行列保持手段で行うことで、受信順序に依存しない処理装置を構成可能となる。また、上記で説明したように信号データ列I受信順序が逆順で送られる場合、行列を逆アドレスで保持することで、回路を変更することなく構

成可能となる。

【0045】上記装置において、信号データ列Iの受信順序の保存を行列保持手段に与える制御データで行うことで受信順序に依存しない処理装置を構成可能となる。例えば、受信順序アドレスをレジスタによるテーブルとして構成することにより、受信順序に非依存及び動的な変更が可能となる。

【0046】上記実施形態においては、回路規模の比較的小さいセクタの数が増えるが、従来技術に比べ、回路規模の比較的大きい加算器の数が減るので、全体としては、回路規模を小さく抑えることができる。

【0047】また、ROMは、検査行列の行列要素が1である部分の行列内での位置のみを格納すればよいので、行列要素を全部記憶する必要が無く、メモリ容量が小さくて済む。更に、LDPCの場合には、行列要素の内、1の数が比較的少ないので、特にメモリ容量の節約に有効である。

【0048】なお、上記実施形態の説明においては、入力データ列Iは、信号値が2値のビット列であることを前提にしたが、必ずしもこれに限定されるものではなく、信号値が各要素の値として実数を取るような場合にも同様に適用可能である。

【0049】LDPCとその符号化については、詳しくは、下記文献を参考にされたい。・和田山正 「低密度パリティ検査符号とその復号方法について」磁気記録研究会発表論文、2001年12月、[http://vega.c.oka-pu.ac.jp/~wadayama/welcome\\_j.html](http://vega.c.oka-pu.ac.jp/~wadayama/welcome_j.html)

・Tadashi Wadayama, "An Extension of Gallager Ensemble of Low Density Parity Check Codes", IEICE TRAN S. FUNDAMENTALS, VOL. E85-A, NO.1 JANUARY 2002

（付記1）行列の行列要素を格納する格納手段と、初期値として全て0である値を格納し、順次行われる演算結果を順次格納するレジスタ手段と、該レジスタ手段から出力される値と入力データ値とを加算する加算手段と、前記行列の行列要素値に基づいて、必要な値を該レジスタ手段から該加算手段に入力し、入力データ値と該レジスタ手段からの値を加算させる演算制御手段と、加算器の出力と、該レジスタ手段の出力を適切に選択して再び該レジスタ手段に格納するループバック手段と、を備える行列演算処理装置。

【0050】（付記2）前記格納手段は、検査行列の行列要素が1となる行列要素の位置のみを情報として格納することを特徴とする付記1に記載の行列演算処理装置。

（付記3）前記演算制御手段は、入力データ値の入力順に合わせて前記格納手段から必要な情報を出力させることを特徴とする付記1に記載の行列演算処理装置。

【0051】（付記4）前記行列は、データの符号化に伴う誤り訂正符号の検査行列であることを特徴とする付記1に記載の行列演算処理装置。

(付記5) 前記行列は、LDPC (Low Density Parity Check) 符号の検査行列であることを特徴とする付記1に記載の行列演算処理装置。

【0052】(付記6) 行列の行列要素を格納する格納ステップと、初期値として全て0である値を格納し、順次行われる演算結果を順次レジスタ手段に格納するレジスタステップと、該レジスタ手段の格納値と入力データ値とを加算する加算ステップと、前記行列の行列要素値に基づいて、必要な値を該レジスタ手段の格納値から該加算ステップに渡し、入力データ値と該格納値を加算させる演算制御ステップと、加算器の出力と、該格納値を適切に選択して再び該レジスタ手段に格納するループバックステップと、を備える行列演算処理方法。

【0053】(付記7) 前記格納ステップでは、検査行列の行列要素が1となる行列要素の位置のみを情報として格納することを特徴とする付記6に記載の行列演算処理方法。

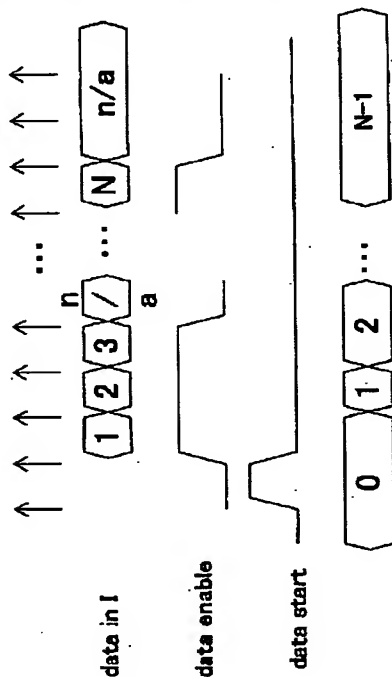
【0054】(付記8) 前記演算制御ステップでは、入力データ値の入力順に合わせて前記格納手段から必要な情報を出力させることを特徴とする付記6に記載の行列演算処理方法。

【0055】(付記9) 前記行列は、データの符号化に伴う誤り訂正符号の検査行列であることを特徴とする付記6に記載の行列演算処理方法。

(付記10) 前記行列は、LDPC (Low Density Pa

【図4】

本発明の実施形態に従った制御手段の実現例を示す図(その2)



arity Check) 符号の検査行列であることを特徴とする付記6に記載の行列演算処理方法。

【0056】

【発明の効果】本発明により、行列演算回路の遅延、回路規模を削減することが可能となる。

【図面の簡単な説明】

【図1】本発明の実施形態に従った行列演算回路の構成例である。

【図2】本発明の実施形態の動作を示す回路例である。

10 【図3】本発明の実施形態に従った制御手段の実現例を示す図(その1)である。

【図4】本発明の実施形態に従った制御手段の実現例を示す図(その2)である。

【図5】本発明の実施形態に従った制御手段の実現例を示す図(その3)である。

【図6】本発明の実施形態に従った制御手段の実現例を示す図(その4)である。

【図7】従来の行列演算回路の例である。

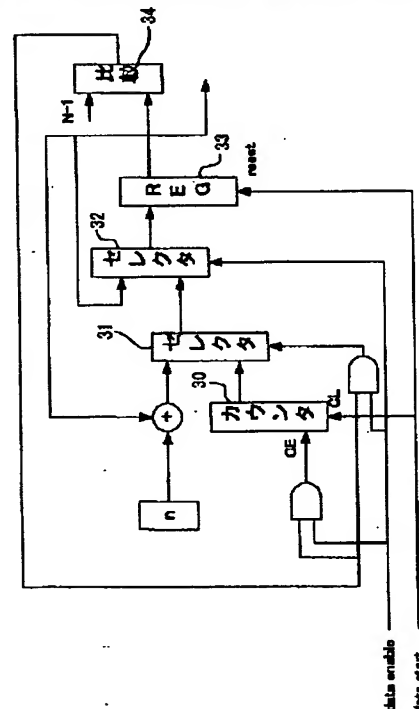
【図8】別の従来例を示す図である。

20 【符号の説明】

- 10 制御手段
- 11 ROM
- 12 reg
- 13 reg (M)

【図5】

本発明の実施形態に従った制御手段の実現例を示す図(その3)

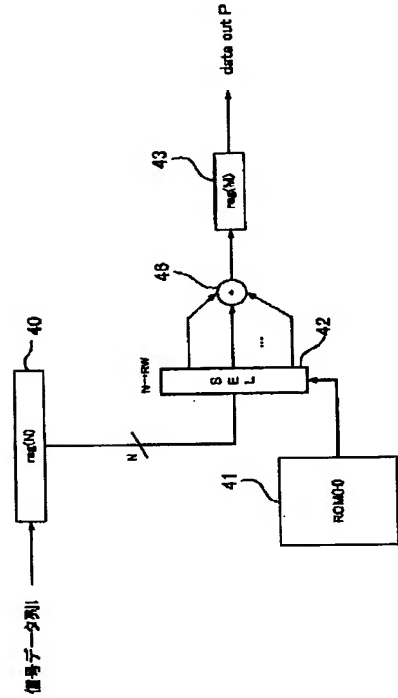
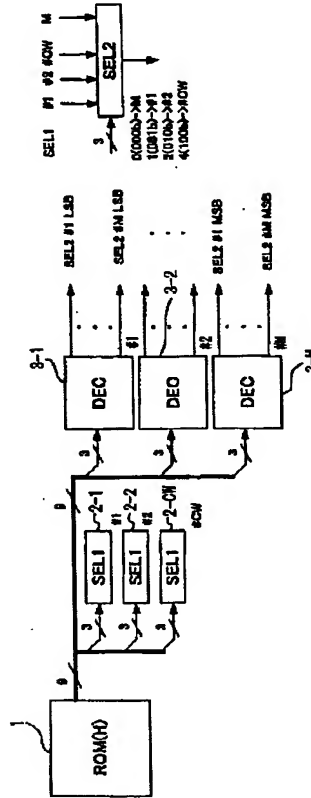
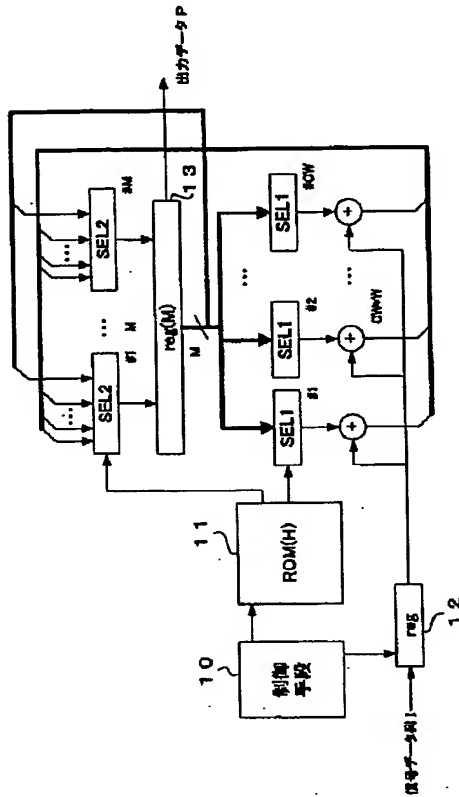


【図1】

【図2】

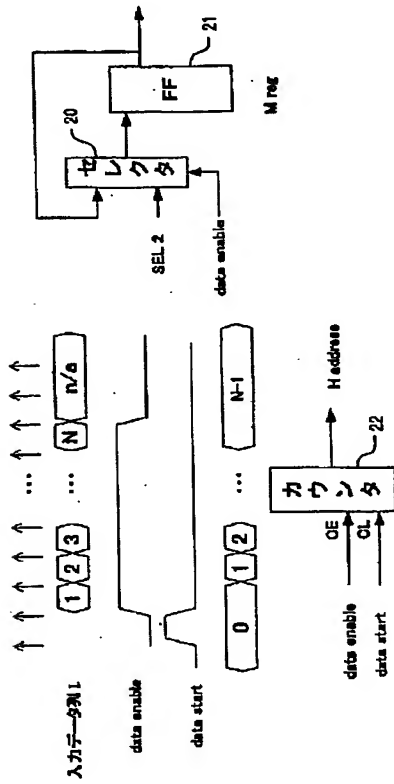
【図7】

本発明の実施形態に従った行列演算回路の構成例 本発明の実施形態の動作を示す回路例 従来の行列演算回路の例



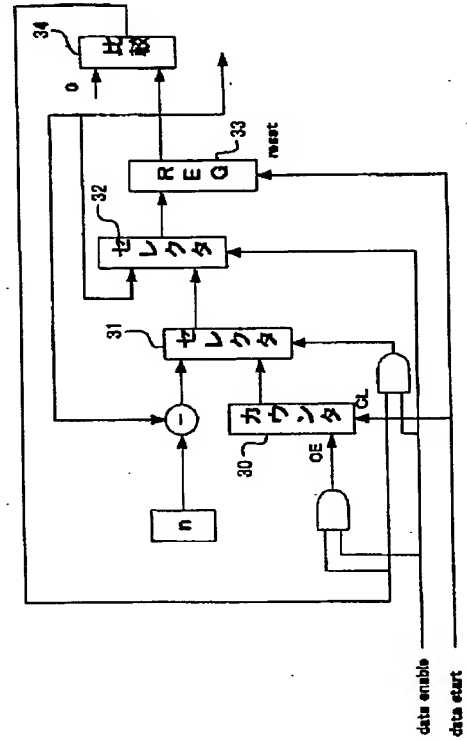
【図 3】

本発明の実施形態に従った制御手段の実現例を示す図（その１）



【図 6】

本発明の実施形態に従った制御手段の実現例を示す図（その４）



【図 8】

別の従来例を示す図

